

10/2/24

WEEK

Program 7 : Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" & derived class "Son" which extends the base class. In Father class, implement a constructor which takes the age & throws the exception WrongAge() when input age  $< 0$ . In Son class, implement a constructor that calls both father & son's age & throws an exception if son's age is  $> =$  father's age.

```
#import java.util.*;
```

```
class WrongAge extends Exception
{
    String msg;
    WrongAge (String msg)
    {
        this.msg = msg;
        System.out.println(msg);
    }
}
```

```
class Father throws WrongAge
{
    int age;
    Father(int age)
    {
        if (age < 0)
            throw new WrongAge("Age can't be negative");
        this.age = age;
    }

    int getAge()
    {
        return age;
    }
}
```

```
class Son extends Father throws WrongAge
{
    int sage;
    Son (int fage, int sage)
    {
        super(fage);
    }
}
```



```

if (fAge <= sAge)
{
    throw new WrongAge("father's age can't be less than son's age");
}

```

```

this.sAge = sAge;
}

```

```

int getSAge()
{
    return sAge;
}

```

```

class AgeTest {

```

```

    public static void main(String args[])
    {

```

```

        Scanner sc = new Scanner(System.in);

```

```

        int b, s;

```

```

        System.out.println("Enter age of father & son");

```

```

        b = sc.nextInt();

```

```

        s = sc.nextInt();

```

```

        try
        {

```

```

            Father obj1 = new Father(b);

```

```

            Son obj2 = new Son(b, s);

```

```

        } catch (WrongAgeException e)
        {

```

```

            System.out.println("Exception : " + e);

```

```

        }
    }
}

```

output :

1) Enter age of father & son : 50 18  
 // no exception



(ii) Enter age of father and son : 18 50  
Father age cannot be less than son's age  
Exception thrown

(iii) Enter age of father and son : -12 20  
Age cannot be negative  
Exception

### Algorithm:-

Step 1: Start

Step 2: Create <sup>user-defined</sup> class Exception <sup>WrongAge</sup> which prints the exception thrown by Father & Son classes

Step 3: Create class Father which <sup>takes</sup> inputs age and checks ~~for~~ if age entered is negative & throws exception

Step 4: Create class Son which extends ~~the~~ Father class, this takes inputs age & sage and checks if age is  $\leq$  sage and throws exception.

Step 5: In main class we <sup>create</sup> ~~create~~ objects for Father & Son and pass user inputs for ages (A try-catch block looks for any exception thrown).

Step 6: Stop.

\* Father & Son classes have a throws exception in their ~~for~~ ~~class~~ declaration (they might throw user defined Exception: WrongAge)

Enter ages of father and son:

50

18

Name: Aditi C

USN: 1BM22CS014

C:\Users\BMSCE\Desktop\014>javac Demo.java

C:\Users\BMSCE\Desktop\014>java Demo

Enter ages of father and son:

18

50

Father age cannot be less than Son's age

Exception

Name: Aditi C

USN: 1BM22CS014

C:\Users\BMSCE\Desktop\014>javac Demo.java

C:\Users\BMSCE\Desktop\014>java Demo

Enter ages of father and son:

-12

20

Age can't be negative

Exception

Name: Aditi C

USN: 1BM22CS014



Program 8: WAP which creates 2 threads, one thread displaying "BMS college of Engineering" once every ten seconds & another displaying "CSE" once every two seconds.

```
#import java.util.*;
```

```
class NewThread implements Runnable
```

```
{
```

```
    String name;
```

```
    Thread t;
```

```
    NewThread(String n) {
```

```
        {
```

```
            name = n;
```

```
            t = new Thread(this, name);
```

```
            t.start();
```

```
        }
```

```
    }
```

```
    public void run() {
```

```
        for (int i = 0; i < 5; i++)
```

```
        {
```

```
            if (name == "one") {
```

```
                System.out.println("BMS College of Engineering");
```

```
                try
```

```
                {
```

```
                    Thread.sleep(10000);
```

```
                }
```

```
            } catch (InterruptedException e) {
```

```
                {
```

```
                    System.out.println("Interrupted");
```

```
                }
```

```
            } System.out.println("BMS College of Engineering");
```

```
        }
```

```
    } else if (name == "two")
```

```
    {
```

```
        System.out.println("CSE");
```

```
        try
```

```
        {
```

```
            Thread.sleep(2000);
```

```
        }
```

```
    } catch (InterruptedException e)
```

```
    {
```

```
        System.out.println("Interrupted");
```

```
    }
```



~~System.out.println ("CSE");~~

```
class Demo {  
    public static void main (String args[])
```

```
    {  
        NewThread obj1 = new NewThread ("One");  
        NewThread obj2 = new NewThread ("Two");
```

```
        try
```

```
        {  
            obj1.t.join();  
            obj2.t.join();
```

```
        }  
        catch (InterruptedException e)
```

```
        {  
            System.out.println ("Main thread interrupted");  
        }  
    }  
}
```

Output :

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

BMS college of Engineering

BMS college of Engineering

BMS college of Engineering

BMS college of Engineering

Algorithm:-



## Algorithm :-

Step 1: Start

Step 2: create a class NewThread that implements the Runnable interface, and takes name of the thread and starts the thread (using start()).

Step 3: the start() method initiates the run() method &

Step 4: in the run method:

for  $i = 0$  to  $k$  do

if (thread name = "one")

output "BMS college of Engineering"

thread sleeps for 10 seconds

else if (thread name = "two")

output "CSE"

thread sleeps for 2 seconds

endif

endfor

Step 5: stop

\* A try and catch block in main function catches an ~~interrupted~~ exception

*Signature*  
16.02.24

```
C:\Users\BMSCE\Desktop\014>javac Test.java
```

```
C:\Users\BMSCE\Desktop\014>java Test
```

```
BMS Ccollege of engineering
```

```
CSE
```

```
CSE
```

```
CSE
```

```
CSE
```

```
CSE
```

```
BMS Ccollege of engineering
```

```
BMS Ccollege of engineering
```

```
BMS Ccollege of engineering
```

```
BMS Ccollege of engineering
```

```
Name: Aditi C
```

```
USN: 1BM22CS014
```